REMARKS/ARGUMENTS

Office Action PP 1: Information Disclosure Statement

Thank you for considering the information disclosure statement submitted November 18, 2005.

Office Action PP 2: The specification

In accordance with the Examiner's objection to the specification, the case has been amended to state "We claim."

Office Action PP3: Objections to the claims

The examiner has made 12 rejections to claim informalities. All 12 items have been corrected.

In paragraph 4 of the Office Action, the Examiner refers to Claim 4 and after reading the claims, we believe the Examiner intended to reference Claim 5; and thus Claim 5 has been amended.

Prior Art Rejections

The Examiner primarily relies on US. Patent 6,717,599 issued to Olano. Applicant respectfully traverses the Examiner's application of Olano because Olano bears no substantive analogy to the claimed inventions.

Olano proposes a system for implementing mathematical derivatives using graphics hardware. The entirety of the Olano disclosure centers around a single method for implementing mathematical derivatives and that method is shown in figure 3. The method specifies: receiving "a data structure that represents the operation of one or more program statements, ***wherein at least one of the program statements requires derivative information***"; marking "a node of the data structure ***that determines derivative information***"; and applying " a transform rule to the node marked in step 320 to obtain a transformed data structure, wherein the operations represented by the transformed data structure can be implemented with graphics hardware interface program." See Figure 4, emphasis added. In accordance with this single method, Olano's compiler transforms program statements into a tree structure comprising nodes that are either single operators or single variables. Olano's system then performs a rule look-up to determine how a

specific tree-segment can be transformed so that the involved derivative math can be performed by resident graphics hardware. Thus, in ▯ ssence, Olan's system merely suggest how to use a low-level tree and rule look-up to perform derivative math on a GPU.

The Examiner also relies heavily on U.S. Patent 6,600,840 issued to McCrossin. Applicant also respectfully traverses the Examiner's application of McCrossin, because like Olano, McCrossin bears no analogy to the claimed inventions.

McCrossin proposes a system for changing the format of an image by applying a series of filters. The filters are applied to the source image for each parameter of the source image that does not match a parameter of the requested or desired result image. In ▯ ssence, McCrossin proposes a system to simplify the conversion of the ***same*** image from one ***format*** to another; for example from a BMP file to a TIFF file. See for example Col. 6 line 40 et. Seq.

Office Action PP 13 – 22: Section 102 rejections

In PP 14, the Examiner has rejected Claim 75 as anticipated by Olano (U.S. Patent 6,717,599). Applicant respectfully traverses this rejection for at least the following exemplary reasons. The Examiner cites the following Olano excerpts to support the proposition that Olan discloses "creating a first graph associated with the first image representation where software routines for creating such graph execute on a CPU":

Column 5, lines 3-10;

> As shown in FIG. 2, compiler **104** comprises a set of derivative transform rules **202** that are applied to an input **204** to produce an output **206**. In an embodiment, input **204** is a text file containing one or more shading language derivative operator statements and/or one or more statements that require derivative information. Input **204** is converted into a tree data structure representation by compiler **104**. The nodes of the tree data structure representation

13. Column 4, lines 30-35; and

> maximum compatibility with graphics systems. Alternatively, compiler **104** can be implemented so that it will transform high level statements into statements designed to take advantage of specialized graphics hardware available in a particular graphics system, thereby optimally transforming high level statements to run on a particular graphics system.

> 14. Column 3, lines 48-53.
>
> tems. In one embodiment of the invention, a compiler transforms shading language statements into graphics API statements that can be implemented with multiple passes through a graphics pipeline. It is a feature of the present invention that it can be implemented on a wide range of available computer graphics systems including, but not limited to, a general purpose computer having a graphics processor (single chip or multiple chip), a specially designed graphics computer, a graphics machine, a parallel processing graphics machine, a graphics card, graphics hardware, a graphics accelerator, a computer network architecture, et cetera.

As is evident from the three excerpts, Olano offers no discussion regarding processing anything on a CPU; quite-the-opposite, Olano focuses on the use of graphics hardware only. In addition, as discussed above and further illustrated here, Olano's trees are not associated with an image representation but rather with program statements. This is clearly and definitively stated in the invention summary:

> The present invention provides a method, system, and computer program product for implementing derivative operators at an interactive rate in computer graphics systems. In an embodiment, ***a data structure representing the operation of one or more computer program statements*** is received by a compiler and transformed into a tree data structure. Nodes of the tree data structure that determine derivative information are marked. A transform rule is applied to the marked nodes to transform the tree data structure into a data structure that can be implemented with graphics hardware interface program statements.
> Olano, Col. 41 – 52.

Regarding claim 75 the Examiner further cites Olano Col. 5 , lines 33-60 to support Olano's anticipation of the following claim elements:

> determining an intersection of the first graph's global domain of definition and global region of interest; resolving a first node in said first graph by running software routines on said CPU to (i) determine if said first node may be combined with a second node and (ii) create program steps for calculating and storing only the portions of any intermediary mage that relate to said intersection.

As is evident from the excerpt from the Examiner's citation (below), Olano proposes nothing even analogous to the claimed limitations above.

> In an embodiment, derivative transform rules **202** comprise a text file containing a plurality of pattern match rules or patterns to match. Matching happens by marking and reducing the data structure of input **204**. The marking (or labeling) of the nodes of input **204** that require derivative information is done bottom-up through the tree using dynamic programming to find a least-cost set of pattern match rules. Reducing input **204** is done top-down, using the pattern match rules, with one semantic action run before a marked node's children are reduced and one run after. The number of passes through a graphics pipeline that are required to implement input **204** is used to represent cost and thus determine a least-cost set of pattern match rules.
>
> Storing pattern match rules in a text file allows the match rules to be readily edited or changed. By changing the match rules, compiler **104** can be quickly and flexibly reconfigured to operate with a different set of high level programming statements or shading language statements or with different graphics hardware. Editing or changing the pattern match rules and the costs determination for compiler **104** enables compiler **104** to transform any set of high level statements, e.g., shading language derivative operators, into graphics API statements (i.e., multiple passes through a graphics pipeline). This embodiment of the invention can take full advantage of any specialized graphics hardware that may be present in a particular graphics system, and thus optimally transforming high level statements for implementation on a particular graphics system.

The Examiner's cited excerpt discusses use of Olano's pattern match rules. As the excerpt states (interpreted as broadly as possible) the pattern match rules merely "transform any set of high level statements . . . into graphics API statements" for the purpose or performing operations on a graphics system. See lines 53 – 60. In fact, neither the cited excerpt, nor anything else in Olano proposes any of the following concepts: a global domain of definition; a global region of interest; the intersection of the foregoing two items; running routines on a CPU to resolve a node; creating program steps relating to the aforementioned intersection. These claimed items are entirely absent in any portion of Olano including the Examiners cited portions.

Finally, with respect to claim 75, the Examiner sites two excerpts (Col. 4, lines 30-35 and Col. 3, lines 48-53) as providing anticipatory disclosure of the claimed portion "the program steps for compilation on the CPU and execution on a GPU."

> Alternatively, compiler **104** can be implemented so that it will transform high level statements into statements designed to take advantage of specialized graphics hardware available in a particular graphics system, thereby optimally transforming high level statements to run on a particular graphics system.

> The present invention provides a method, system, and computer program product for implementing derivative operators at an interactive rate in computer graphics systems. In one embodiment of the invention, a compiler transforms shading language statements into graphics API statements that can be implemented with multiple passes through a graphics pipeline. It is a feature of the present invention that it can be implemented on a wide range of available computer graphics systems including, but not limited to, a general purpose computer having a graphics processor (single chip or multiple chip), a specially designed graphics computer, a graphics machine, a parallel processing graphics machine, a graphics card, graphics hardware, a graphics accelerator, a computer network architecture, et cetera.

As is evident from the cited excerpts, Olano's focus and only disclosure relates to processing by graphics systems of one kind or another. Neither the cited excerpts nor anything else in Olano discusses any division of work between CPU and GPU, much less a division of work as claimed (the claimed program steps being compiled on the CPU and executed on the GPU)

For all the reasons set forth above, claim 75 is clearly allowable over the cited art and thus currently in condition for allowance.

In Office Action PP 15 – 22, the Examiner rejects claims 79 – 84 as being anticipated by McCrossin. As discussed earlier, McCrossin proposes a system for changing the format of an image by applying a series of filters. In essence, McCrossin proposes a system to simplify the conversion of the ***same*** image from one ***format*** to another; for example from a BMP file to a TIFF file. See for example Col. 6 line 40 et. Seq. Thus, the API discussed by McCrossin is not an image processing API as set forth

in claim 79 but rather, as explicitly stated in the Examiner's cited excerpt, a "read" and "write" API. See Col. 6, lines 20-25 quoted below.

> image data in the format specified by application **101**. A request to read and write image data is accomplished by employing read and write application program interface (API) calls, read API calls **129** and write API calls **131** that are sent to transform object **103** in accordance with a preferred embodiment of the present invention.

Referring to McCrossin's figure 6, when the transform object 103 receives a ***read*** or ***write*** API call, it helps fulfill that job by providing a behind-the-scenes filter system for getting the correct file format as listed at item 137. Critically, the API is to ***read*** or ***write***, ***not*** to perform image processing as claimed. Furthermore, since the McCrossin API is for read and write operations, McCrossin does not and can not disclose a bundle of API services as claimed; i.e. API services related to filter objects, API services related to image objects, API services related to context objects and API services related to vector objects. Such a bundle of API services as claimed is entirely absent from McCrossin at least because a major purpose of McCrossin is to isolate the application layer from the task of doing a file format conversion (e.g. the McCrossin application does NOT use an API to call all "filters" involved in the file format conversion). Thus, in view of this clarification, claim 79 should be in condition for allowance and reconsideration is respectfully requested. Claims 80 through 83 depend from claim 79 and therefore should be allowable for the same reasons.

Similar to claim 79, the Examiner asserts that claim 84 is invalid over McCrossin. Applicant traverses this rejection because, as explained above McCrossin discloses only read and write APIs and does not and can not disclose a bundle of API functions as claimed; i.e. API functions to create image objects; API functions to create context objects; API functions to create filter objects; API functions to set filter object parameters; API functions to obtain filter object output; and API functions to convert image objects to context objects. Thus, in view of this clarification, claim 84 should be in condition for allowance and reconsideration is respectfully requested.

6. Office Action PP 23 – 89: Section 103 Rejections

The Examiner has rejected claims 1, 2, 4-6, 11-23, 25-32, 36-41, 43-47, 51-56, 58, 61-64, 66-74 and 85 under Section 103 in view of the combination of Olano and McCrossin. Applicant respectfully traverses this rejection.

As explained above, Olano is essentially inapplicable to the claimed inventions because Olano proposes a system for implementing mathematical derivatives using graphics hardware. The entirety of the Olano disclosure centers around a single method for implementing mathematical derivatives and that method is shown in figure 3. The method is described and cited above in this response.

In Office Action PP 26, the Examiner rejects claim 1 and cites Olano (Col. 6, line 50 – Col. 7, line 3) for the allegedly anticipatory proposition that Olano discloses "a first process requesting a filter from a second process; the related filter and initial image comprising a program." However, Olano discloses nothing of sort, focusing only on the manipulation of program code so that graphics hardware can perform mathematical derivatives. The Olano portion cited by the Examiner discusses estimating the size of an anti-aliasing filter; there is no discussion regarding a request for filters between processes. Thus, other than mentioning a filter, the Examiner's citation has no reference or relevance to the cited claim language requiring a first process to request a filter from a second process. The Olano citation is as follows:

> the u and v parametric directions. For example, it is necessary to compute derivative values for parametric patch primitive **400** in both the u and v parametric directions in order to estimate the size of a filter size needed to antialias a texture applied to parametric patch primitive **400** or to determine the normal to small surface area **402**. These derivative values are typically obtained using shading language statements that implement derivative operators.
>
> A typical shading language statement that implements derivative operators is a statement that returns a differential area value at a point p. The square root of a differential area is a length, which can be used as an approximation of the amount that a point p changes between adjacent points in a texture. This shading language statement, denoted herein as area(p), is commonly used to estimate the size of a filter that is needed to antialias a texture.
>
> The following shading language procedure or shader (expressed in pseudo-code), for example, would return a
>
> **7**
>
> differential surface area at a point p, which could then be used to estimate the size of a filter that is needed to antialias a texture:

With respect to claim 1 the Examiner further puts forth that Olano (at Col. 2, lines 52-56) shows the steps of "said second process compiling the program, yielding a compiled program; running at least a portion of the complied program to apply a function of the filter to the image, yielding an pixel-image result." The cited quotation is as follows:

> graphics hardware interface program statements. In one embodiment of the invention, the compiler transforms shading language statements into graphics application programming interface statements that can be implemented with multiple passes through a graphics pipeline. It is a feature of

It is quite evident that neither the cited portion nor anything in Olano discloses any of the following: which among a plurality of processes compiles the claimed program; or running the claimed compiled program to yield a pixel-image result." Since Olano

speaks only to performing mathematical operations, it can not have any relevance to yielding a pixel-image result.

With respect to claim 1, the Examiner further asserts that McCrossin teaches "editing an initial image, comprising the steps of a first process requesting a filter from a second process, said first process defining a relationship between the filter and the initial image." However, the Examiner's application of McCrossin is misplaced because McCrossin's transform object "selects" a filter from a library and "employs" that filter; it does not request a filter from another process or deliver a filter in accordance with the request of another process. See Col 6, line 46 – Col. 7, line 9, which is the examiner's citation copied directly below.

> API read calls **133** and API write calls **135** are calls to the image readers and writers to control the reading or writing of image data. The API calls employed are defined by the requirements of each particular reader or writer within image readers/writers **137** in accordance with a preferred embodiment of the present invention. Transform object **103** determines an actual image vector **139**, which describes the format of the image to be read or written. Image request vector **127** is received from application **101** and is compared to actual image vector **139** to determine what filters are needed. After such an evaluation, filter stack **141** is constructed to return the image data specified in image request vector **127**.
>
> Filters are accessed by transformation object **103** from filter library **143**, as depicted in FIG. 7. Filter library **143** contains a number of different filters available for performing various image transformations. In the depicted example, filter library **143** includes the following filters: photometric **143***a*, rotate **143***b*, crop **143***c*, pad **143***d*, scale **143***e*, bit pad **143***f*, dither **143***g*, gray **143***h*, color transform **143***i*, decode **143***j*, and encode **143***k*. These filters are employed by transformation object **103** to provide the necessary transfor-

7

mation or alteration of image data to return image data in a form as specified in image request vector **127**. Transform object **103** selects a number of filters to create a filter stack **141** for use in manipulating the image data. These filters may be used when image readers/writers **137** is employed to read or write data in accordance with a preferred embodiment of the present invention. More information on filters may be found in Foley et al., *Computer Graphics: Principles and Practice* (2d ed. 1991).

Thus McCrossin simply does not apply to the claim language.

Finally, with respect to claim 1, Applicant traverses the Examiner's contention that there is any motivation to combine Olano and McCrossin. Olano is a system for doing derivative math in graphics hardware and McCrossin is a way to change file formats. There is no reason combine references that differ as much as these.

Thus, in view of Applicant's arguments, claim 1 should be allowable and the Examiner's reconsideration is respectfully requested. Furthermore, since claims 2, 4-6, 8, and 11-21 all depend from claim 1, those claims should likewise be allowable and reconsideration is respectfully requested.

In Office Action PP 42, the Examiner also rejects independent claim 22 over the combination of Olano and McCrossin. Applicant traverses this rejection generally for the reasons stated above and specifically as follows. Furthermore, without waiving argument on all the limitations, Applicant points out the following failures of Olano and McCrossin that should be easy to agree upon. The Examiner cites Olano's Col. 4, lines 11-21 (see below) as allegedly disclosing "the filter requested by the first process."

> The application program layer comprises application **102**. Application **102** represents any high level application. For example, application **102** might be a program for rendering computer graphics imagery to be used in a movie, or a program for visualization of scientific data. Application **102** will typically allow a user, for example, a movie technical director or a scientist, to describe images or scenes using a high level programming language or a shading language, which causes object files residing at the graphics toolkit layer and/or the graphics API layer to be executed by the graphics system embodying architecture **100**.

As is evident from the Olano portion cited by the Examiner (quoted just above), Olano does not anywhere disclose a first process (or any process) requesting a filter as claimed.

In addition, the Examiner cites the same points in Olano for the claimed first and second microprocessor (Olano Col. 4, lines 11-35 and Col. 4, lines 30-35). While, Col. 4 of Olano may indeed list a variety of exemplary hardware, nothing in Col. 4 or elsewhere teaches to use a first and second microprocessor according to the claimed division of work. As explained earlier, Olano teaches the use of graphics hardware to perform derivative mathematics. The only attribution of work even implied is that the graphics hardware will do the math. Even the most expansive reading of Olano can not imply, as claimed, a first microprocessor running two interrelated processes where one of those processes causes the creation of a specifically claimed data structure; and then a second microprocessor for running a program created by using the specifically claimed data structure. To the contrary, Olano discusses converting program lines into a node tree and then processing the node tree so that graphics hardware can be used to do derivative math. Olano simply lacks at least both the specifically claimed data structure and the expressly claimed division of work between microprocessors.

Furthermore, the examiner alleges that Olano's Col. 11, lines 14-18 discloses a memory for storing a pixel image resulting from running a specifically claimed program. And while that section does discuss a frame buffer as illustrative hardware, it cannot be, as claimed, a buffer for storing a pixel image resulting from running the claimed program. This is because the only "programs" arguing taught by Olano are for doing

math, not for making or editing images. Thus, Olano does not and can not suggest making or editing images using the specifically claimed program.

Moreover, with respect to claim 22, the Examiner asserts that McCrossin teaches that a data structure comprises a relationship between an image and a filter. While the Examiner's cited excerpt (Col 6, line 46 – Col. 7, line 9) does discuss "filters" and "images," it most certainly does NOT discuss any data structure, much less the specifically claimed data structure that comprising the claimed relationship between an initial image and a specific filter. See Col 6, line 46 – Col. 7, line 9, which is the examiner's citation copied directly below.

> API read calls **133** and API write calls **135** are calls to the image readers and writers to control the reading or writing of image data. The API calls employed are defined by the requirements of each particular reader or writer within image readers/writers **137** in accordance with a preferred embodiment of the present invention. Transform object **103** determines an actual image vector **139**, which describes the format of the image to be read or written. Image request vector **127** is received from application **101** and is compared to actual image vector **139** to determine what filters are needed. After such an evaluation, filter stack **141** is constructed to return the image data specified in image request vector **127**.
>
> Filters are accessed by transformation object **103** from filter library **143**, as depicted in FIG. **7**. Filter library **143** contains a number of different filters available for performing various image transformations. In the depicted example, filter library **143** includes the following filters: photometric **143***a*, rotate **143***b*, crop **143***c*, pad **143***d*, scale **143***e*, bit pad **143***f*, dither **143***g*, gray **143***h*, color transform **143***i*, decode **143***j*, and encode **143***k*. These filters are employed by transformation object **103** to provide the necessary transfor-

> 7
>
> mation or alteration of image data to return image data in a form as specified in image request vector **127**. Transform object **103** selects a number of filters to create a filter stack **141** for use in manipulating the image data. These filters may be used when image readers/writers **137** is employed to read or write data in accordance with a preferred embodiment of the present invention. More information on filters may be found in Foley et al., *Computer Graphics: Principles and Practice* (2d ed. 1991).

Thus, given the lack of any disclosure regarding a data structure comprising the relationship information, as claimed: McCrossin simply does not apply to the claim language.

Finally, with respect to claim 22, Applicant traverses the Examiner's contention that there is any motivation to combine Olano and McCrossin. Olano is a system for doing derivative math in graphics hardware and McCrossin is a way to change formats. There is no reason combine references that differ as much as these.

Thus, in view of Applicant's arguments, claim 22 should be allowable and the Examiner's reconsideration is respectfully requested. Furthermore, since claims 23, and 25 - 27 all depend from claim 22, those claims should likewise be allowable and reconsideration is respectfully requested.

In Office Action PP 47, the Examiner also rejects independent claim 28 over the combination of Olano and McCrossin. Applicant traverses this rejection generally for the reasons stated above and specifically as follows. Furthermore, without waiving argument on all the limitations, Applicant points out the following failures of Olano and McCrossin that should be easy to agree upon. The Examiner cites Olano's Col. 2, lines 45-56 and Col. 4, lines 30-35 (see below) as allegedly disclosing "a second process servicing the requests of the first process, the servicing comprising the steps of optimizing a graph representing the result image; compiling the optimized graph; causing the rendering of the complied optimized graph."

> tems. In an embodiment, a data structure representing the operation of one or more computer program statements is received by a compiler and transformed into a tree data structure. Nodes of the tree data structure that determine derivative information are marked. A transform rule is applied to the marked nodes to transform the tree data structure into a data structure that can be implemented with graphics hardware interface program statements. In one embodiment of the invention, the compiler transforms shading language statements into graphics application programming interface statements that can be implemented with multiple passes through a graphics pipeline. It is a feature of

> Alternatively, compiler **104** can be implemented so that it will transform high level statements into statements designed to take advantage of specialized graphics hardware available in a particular graphics system, thereby optimally transforming high level statements to run on a particular graphics system.

As is evident from the Examiner's citations quoted above, Olano merely proposes making a node tree from program statements and then adapting the tree to perform mathematical derivatives on graphics hardware. Olano doesn't even purport to do so in the context of a second process servicing the request of a first process and Olano doesn't mention or suggest optimizing anything (Olano's adaptation of the node tree for math and the graphics processor is not and does not purport to be anything like optimization).

The Examiner also cites McCrossin (Col. 5, lines 49-50, Col. 6, lines 51-58, and lines 9-21) to allegedly anticipate a first process requesting creation of a context, the first process indicating parameters associated with the creation of the result image; and the first process requesting that the result image be rendered to the context. However, the cited excerpts merely discuss McCrossin's transform object selection of vectors for McCrossin's format conversions. This has nothing to do with creating contexts and rendering to same. McCrossin is focused on the conversion of formats and thus is totally unconcerned with the notion of context creation or use, much less in the specifically claimed manner.

Finally, with respect to claim 28, Applicant traverses the Examiner's contention that there is any motivation to combine Olano and McCrossin. Olano is a system for doing derivative math in graphics hardware and McCrossin is a way to change formats. There is no reason combine references that differ as much as these.

Thus, in view of Applicant's arguments, claim 28 should be allowable and the Examiner's reconsideration is respectfully requested. Furthermore, since claims 29-32 and 36-40 all depend from claim 28, those claims should likewise be allowable and reconsideration is respectfully requested.

In Office Action PP 55, the Examiner rejects claim 43 with citation back to the Examiner's arguments for claim 28; except the examiner suggests that claim 43 has the additional limitation of a first process and a second process running on a first microprocessor and rendering occurring on a second microprocessor. The Examiner alleges anticipation of this limitation citing Olano at Col.3, lines 48-53 and Col. 4, lines 30-35; both excerpts copied directly below.

> tems. In one embodiment of the invention, a compiler transforms shading language statements into graphics API statements that can be implemented with multiple passes through a graphics pipeline. It is a feature of the present invention that it can be implemented on a wide range of available computer graphics systems including, but not limited to, a general purpose computer having a graphics processor (single chip or multiple chip), a specially designed graphics computer, a graphics machine, a parallel processing graphics machine, a graphics card, graphics hardware, a graphics accelerator, a computer network architecture, et cetera.

> maximum compatibility with graphics systems. Alternatively, compiler **104** can be implemented so that it will transform high level statements into statements designed to take advantage of specialized graphics hardware available in a particular graphics system, thereby optimally transforming high level statements to run on a particular graphics system.

No reasonable reading of these excerpts can find any discussion or teaching regarding the claimed division of work between two microprocessors. These passages do not even suggest any specific division of work between processors of any kind, much less the specifically claimed division of work.

Furthermore, with respect to claim 43, Applicant refers to arguments made previously with respect to claim 28. Thus, in view of Applicant's arguments, claim 43 should be allowable and the Examiner's reconsideration is respectfully requested. Furthermore, since claims 41-47 and 51-56, those claims should likewise be allowable and reconsideration is respectfully requested.

In Office Action PP 57, the Examiner also rejects independent claim 58 over the combination of Olano and McCrossin. Applicant traverses this rejection generally for the reasons stated above and specifically as follows. Furthermore, without waiving argument on all the limitations, Applicant points out the following failures of Olano and McCrossin that should be easy to agree upon. The Examiner cites Olano's Col. 10, lines 59-65 (see below) as allegedly disclosing an API call for the function of creating an image.

> Rasterization module **814** combines the output of vertex operation module **812** and pixel operation module **820** to form image fragments. Image fragments correspond to pixels in frame buffer **818**. Image fragments have both color values, such as red, green, blue, and alpha, and a depth value, as would be known to a person skilled in the relevant art.

As is evident from the cited excerpt reproduced above, there is no teaching or suggestion of for an API call having the function to create an image. Furthermore, while Olano may propose the concept of an image it does NOT propose, teach or suggest the claimed step of ***associating*** an image or anything else with a with the claimed function call. Once again, Olano is a system to perform math with graphics hardware, thus Applicant's specifically crafted claims regarding a high-level graphics API are far removed from anything in the Olano disclosure. Furthermore, the Examiner's citation to McCrossin is

similarly misplaced because, as explained earlier, McCrossin's only APIs are to read and write and thus McCrossin offers nothing in terms of a graphics API, much less defining relationships between such APIs and the specifically claimed objects.

Finally, with respect to claim 58, Applicant traverses the Examiner's contention that there is any motivation to combine Olano and McCrossin. Olano is a system for doing derivative math in graphics hardware and McCrossin is a way to change file formats. There is no reason combine references that differ as much as these.

Thus, in view of Applicant's arguments, claim 58 should be allowable and the Examiner's reconsideration is respectfully requested. Furthermore, since claims 61-62 depend from claim 58, those claims should likewise be allowable and reconsideration is respectfully requested.

In Office Action PP 60, the Examiner has rejected claim 63, 64, 66, and 67 making only reference to the Examiner's arguments with respect to claims 58, 59, 61 and 62. Applicant traverses the Examiner's rejection similarly referring back and forward in this paper to Applicant's respective arguments.

Finally, with respect to independent claim 63, Applicant traverses the Examiner's contention that there is any motivation to combine Olano and McCrossin. Olano is a system for doing derivative math in graphics hardware and McCrossin is a way to change file formats. There is no reason combine references that differ as much as these.

Thus, in view of Applicant's arguments, claim 63 should be allowable and the Examiner's reconsideration is respectfully requested. Furthermore, since claims 64, 66-73 all depend from claim 63, those claims should likewise be allowable and reconsideration is respectfully requested

In Office Action PP 67, the Examiner also rejects independent claim 74 over the combination of Olano and McCrossin. Applicant traverses this rejection generally for the reasons stated above and specifically as follows. Furthermore, without waiving argument on all the limitations, Applicant points out the following failures of Olano and McCrossin that should be easy to agree upon. As discussed above, Olano certainly does not disclose optimizing a graph representing an image and the Examiner's citation to Col. 4, lines 30-36 does not even suggest same:

> maximum compatibility with graphics systems. Alternatively, compiler **104** can be implemented so that it will transform high level statements into statements designed to take advantage of specialized graphics hardware available in a particular graphics system, thereby optimally transforming high level statements to run on a particular graphics system.

Furthermore, Olano does not disclose any division of work between processing entities, much less the specifically claimed division of work between a CPU and GPU. The Examiner has cited the same excerpt directly above for this anticipation allegation; and no reasonable reading to the passage or anything else in Olano can infer the claimed division of work. Furthermore, the Examiner cites McCrossin for alleged anticipation regarding rendering to a specified context. However, the cited excerpt merely discusses McCrossin's transform object's selection of vectors for McCrossin's format conversions. This has nothing to do with creating contexts and rendering to contexts. McCrossin is focused on the conversion of formats and thus is totally unconcerned with the notion of context creation or use, much less in the specifically claimed manner.

Finally, with respect to claim 74, Applicant traverses the Examiner's contention that there is any motivation to combine Olano and McCrossin. Olano is a system for doing derivative math in graphics hardware and McCrossin is a way to change formats. There is no reason combine references that differ as much as these.

Thus, in view of Applicant's arguments, claim 74 should be allowable and the Examiner's reconsideration is respectfully requested.

In Office Action PP 68, the Examiner rejects claim 85. Applicant traverses this rejection because claim 85 depends upon other claims which have been demonstrated to overcome the cited art. Thus, claim 85 should be in condition for allowance and reconsideration is respectfully requested.

In Office Action PP 69, the Examiner rejects claims 3, 33-35, and 48-50 as being unpatentable over the 3-way combination of Olano, McCrossin and Levy (US 20020033844A1). Applicant traverses this rejection because: claim 3 depends from claim 1 which has been demonstrated to overcome the cited art; claims 33-35 depend

from claim 28, which has been demonstrated to overcome the cited art; and claims 48-50 depend from claim 43, which has been demonstrated to overcome the cited art.

Finally, with respect to claims 3, 33-35, and 48-50, Applicant traverses the Examiner's contention that there is any motivation to combine these three widely ranging references. There is no reason combine references that differ as much as these.

Thus, claims 3, 33-35 and 48-50 should be in condition for allowance and reconsideration is respectfully requested.

In Office Action PP 75, the Examiner rejects claims 7, 9, 60 and 65 as being unpatentable over the 3-way combination of Olano, McCrossin and Parikh (US 6411301). Applicant traverses this rejection because: claims 7 and 9 depends from claim 1 which has been demonstrated to overcome the cited art; claims 60 depends from claim 58, which has been demonstrated to overcome the cited art; and claim 65 depends from claim 63, which has been demonstrated to overcome the cited art.

Finally, with respect to claims 7, 9, 60 and 65, Applicant traverses the Examiner's contention that there is any motivation to combine these three widely ranging references. There is no reason combine references that differ as much as these.

Thus, claims 7, 9, 60 and 65 should be in condition for allowance and reconsideration is respectfully requested.

In Office Action PP 80, the Examiner rejects claim 10 as being unpatentable over the 4-way combination of Olano, McCrossin, Parikh and Doyle (US 6867779). Applicant traverses this rejection because claim 10 depends from claim 1 which has been demonstrated to overcome the cited art.

Finally, with respect to claim 10, Applicant traverses the Examiner's contention that there is any motivation to combine these four widely ranging references. There is no reason combine references that differ as much as these and it is simply inappropriate and counter legal precedent to combine four references that are not very tightly coupled together for purposes of motivation to combine.

Thus, claims 10 should be in condition for allowance and reconsideration is respectfully requested.

In Office Action PP 81, the Examiner rejects claim 24 as being unpatentable over the 3-way combination of Olano, McCrossin and Sturges (US 5854637). Applicant traverses this rejection because claim 24 depends from claim 22, which has been demonstrated to overcome the cited art.

Finally, with respect to claim 24, Applicant traverses the Examiner's contention that there is any motivation to combine these three widely ranging references. There is no reason combine references that differ as much as these.

Thus, claim 24 should be in condition for allowance and reconsideration is respectfully requested.

In Office Action PP 82, the Examiner rejects claims 42 and 57 as being unpatentable over the 3-way combination of Olano, McCrossin and Stokes (US 6977661). Applicant traverses this rejection because: claim 42 depends from claim 28 which has been demonstrated to overcome the cited art; and claims 57 depends from claim 43, which has been demonstrated to overcome the cited art.

Finally, with respect to claims 42 and 57, Applicant traverses the Examiner's contention that there is any motivation to combine these three widely ranging references. There is no reason combine references that differ as much as these.

Thus, claims 42 and 57 should be in condition for allowance and reconsideration is respectfully requested.

In Office Action PP 85, the Examiner rejects claim 59 as being unpatentable over the 3-way combination of Olano, McCrossin and Doyle. Applicant traverses this rejection because claim 59 depends from claim 58, which has been demonstrated to overcome the cited art.

Finally, with respect to claim 59, Applicant traverses the Examiner's contention that there is any motivation to combine these three widely ranging references. There is no reason combine references that differ as much as these.

Thus, claim 59 should be in condition for allowance and reconsideration is respectfully requested.

In Office Action PP 86, the Examiner rejects claims 76 and 77 as being unpatentable over the combination of Olano and Levy (US 20020033844A1) Applicant

traverses this rejection because claims 76 and 77 depend from claim 75 which has been demonstrated to overcome the cited art.

Finally, with respect to claims 76 and 77, Applicant traverses the Examiner's contention that there is any motivation to combine these two widely ranging references. There is no reason combine references that differ as much as these.

Thus, claims 76 and 77 should be in condition for allowance and reconsideration is respectfully requested.

In Office Action PP 89, the Examiner rejects claim 78 as being unpatentable over the combination of Olano and French (US 6266053) Applicant traverses this rejection because claim 78 depends from claim 75 which has been demonstrated to overcome the cited art.

Finally, with respect to claim 75, Applicant traverses the Examiner's contention that there is any motivation to combine these two widely ranging references. There is no reason combine references that differ as much as these.

Thus, claim 75 should be in condition for allowance and reconsideration is respectfully requested.

7. Conclusion

In summary, Applicant traverses all of the Examiner's art-based rejections because the primary references used by the Examiner are plainly inapplicable to the inventions claimed by applicant. Furthermore, in addition to alleging the application of distant art, the Examiner's specific allegations regarding each claim do not incorporate the claimed connectivity between claim elements. For example, with respect to Claim 22, the Examiner has alleged McCrossin teaches that a data structure comprises a relationship between an image and a filter. While the Examiner's cited excerpt (Col 6, line 46 – Col. 7, line 9) does discuss "***filters***" and "***images***," it most certainly does NOT discuss any ***data structure comprising a relationship between an initial image and a specific filter.***

Finally, with regard to all the Section 103 rejections, the Examiner has failed to show any motivation to combine references. For example, the Examiner's two primary

references are Olano and McCrossin. Olano relates to system for using graphics hardware to perform derivative math and McCrossin relates to using the sequential application filters to change the file format of an image (e.g. from BMP to JPG, etc.). A skilled artisan working in the field of the Applicant's claims would not likely be inclined to find one of these references; but even if one were found accidentally, since the topic of the other is so far removed, there would be motivation to find or combine the two. In essence, the combination of Olano and McCrossin "teaches" the non-sensical technical point of graphics systems that convert file formats in order to do derivative calculus.

Having demonstrated distinction from the cited art and corrected informalities as requested, Applicant believes that all the claims of this case are in condition for allowance. If the Examiner has questions or wishes to discuss this matter, please call the undersigned at 832-446-2415.

Respectfully submitted,

Lou Brucculeri, Reg. No. 38,834

Wong, Cabello, Lutsch,
Rutherford & Brucculeri, L.L.P.
20333 SH 249. Suite 600
Houston, TX 77070
832/446-2400